Real-Time Exploration of Multimedia Collections

Juraj Moško, Tomáš Skopal, Tomáš Bartoš, and Jakub Lokoč

Charles University in Prague, Faculty of Mathematics and Physics, SIRET Research Group
Malostranské nám. 25, 118 00 Prague, Czech Republic
`{mosko,skopal,bartos,lokoc}@ksi.mff.cuni.cz`
`http://www.siret.cz`

Abstract. With the huge expansion of smart devices and mobile applications, the ordinary users are consistently changing the conventional similarity search model. The users want to explore the multimedia data, so the typical query-by-example principle and the well-known keyword searching have become just a part of more complex retrieval processes. The emerging multimedia exploration systems with robust back-end retrieval system based on state of the art similarity search techniques provide a good solution. They enable interactive exploration process and implement exploration queries tightly connected with the user interface. However, they do not consider larger response times that might occur. To overcome this, we propose a scalable exploration system **RTExp** that allows evaluating the similarity queries in the near real time depending on user preferences (speed / precision). We describe building parts of the system and discuss various real-time characteristics for the exploration process. Also we provide results from the experimental evaluation of time-limited similarity queries and corresponding exploration operations.

1 Introduction

The continuously growing multimedia collections provide generally unstructured data, so it is almost impossible to completely understand what kind of information these datasets contain and which data might be relevant for individual users. Since it is difficult to index such data or to query it using the traditional approaches (e.g., full-text indexes), new techniques employing the *similarity search* paradigm [21] have emerged.

The similarity search concept applies the content-based comparison of multimedia objects driven by object similarities. This approach typically leads to *query-by-example* searching in which a user specifies the query with an initial object with additional conditions. However, this type of *targeted search* (or controlled querying), with a sample query object is not intuitive enough when compared to real user expectations.

In specific scenarios, the user does not really know what (s)he is searching for, so (s)he cannot provide the appropriate query object. Instead, the user iteratively *browses* or *explores* the database until spotting the right object(s). Here, we talk about *indirect searching*. Recently, there appear many approaches to the interactive indirect search for multimedia collections with the main focus put on *multimedia exploration systems* [14,17], that aims at more sophisticated and faster kind of browsing.

We consider the *exploration* as a user-controlled (interactive) process of viewing and browsing the multimedia collections in a way which is arbitrary, unpredictable, and not

H. Wang and M.A. Sharaf (Eds.): ADC 2014, LNCS 8506, pp. 198–205, 2014.

predefined at all. Initially, the system displays and visualizes a very small portion of the underlying dataset. The user can pick an object, click on or zoom in/out specific parts of the currently visualized items, so that the search becomes partly targeted during the interactive search process.

The recently proposed engines [14,17,2] encounter real problems with processing very large databases (millions of objects or more), because they do not consider the most important challenge – to guarantee the worst query performance for arbitrarily large collections which is the main objective of this paper. Unlike targeted querying, in which the user accepts short delays for the query execution, during exploration the process has to be smooth and instant. Hence, for multimedia exploration the underlying routines have to operate in near real time to provide the illusion of a smooth process.

2 Related Work and Motivation

The concept of multimedia exploration systems has been recently introduced as the combination of two major factors: (1) browsing multimedia collections, and (2) using approximate similarity search [2,14,17]. Each of the systems introduces some novel ideas such as the intuitive interface to a visualized collection [17], or the separated middle layer in the system architecture [16].

The approximate similarity search arises from the idea of exchanging the query effectiveness (precision) for the query efficiency (speed). Users adjust this trade-off at/before the query time using various approximation parameters, for example a probability of an error in the query result [22,6], the maximum number of similarity/distance computations allowed, or the threshold on the improvement of the query result [22].

The classification of approximate search scenarios [15] reveals different principles and organizes them in a consistent way. One of the inspiring approaches is the RC_{ES} which stands for *reducing comparisons* and *early termination*. Another relevant technique is the concept of incremental similarity search [20]. It works with tree index structures and assigns higher priority to more promising nodes. Based on this priority, we traverse unprocessed nodes within the tree and stop the process whenever the most promising node in the processing queue guarantees worse values than currently the "worst" node in the result set (e.g., smaller similarity than the kth nearest neighbor). This technique was later proven to be range-optimal [11] and also optimal for disk page accesses [4]. One of the first adoptions of this idea has been in spatial databases [10] and afterwards also in metric indexes [11].

Although we could utilize any of these systems, they have to be adopted for different purpose (multimedia exploration) than originally designed (similarity queries). As the crucial requirement on the multimedia exploration process is a smooth navigation in the collection, the indexes behind providing partial similarity queries have to guarantee real-time responses, either given a real-time constraint for execution, or even anytime query termination when no time constraint is available. If such a guarantee is satisfied, the entire exploration system could be designed to guarantee smooth navigation of the user in the collection, where the underlying streams of similarity queries are scheduled and executed such that the total real-time constraint is kept.

2.1 Multimedia Exploration in Mobile Devices

There is no doubt that classical computers and notebooks are being dominated by small, smart, touch-based mobile devices constantly connected to the Internet. As a consequence, users can search, browse, download/upload multimedia data practically anytime and anywhere. On the other side, small displays and limited control options represent new challenges for mobile multimedia applications trying to provide the multimedia data in the most convenient and also entertaining way such as multimedia exploration can offer [2,18].

Although multimedia exploration techniques apply index support and approximate search to improve the efficiency of whole process, to the best of our knowledge, none of the related methods considers the real-time responses of the exploration process in the presence of huge multimedia datasets and mobile environment. We believe that guaranteeing a real-time response is one of the most crucial properties of a successful multimedia exploration system, even more important than the steadily high precision of the retrieval. Hence, we focus on approximate search techniques guaranteeing user-defined response time and allowing occasional inaccuracies for long running queries. Furthermore, the response time can be controlled by users who decide the trade-off between the precision and the efficiency. For example, for a casual exploration of an unknown collection the user prefers fast response times, but when searching for some specific objects, the user will wait longer for a more qualitative result.

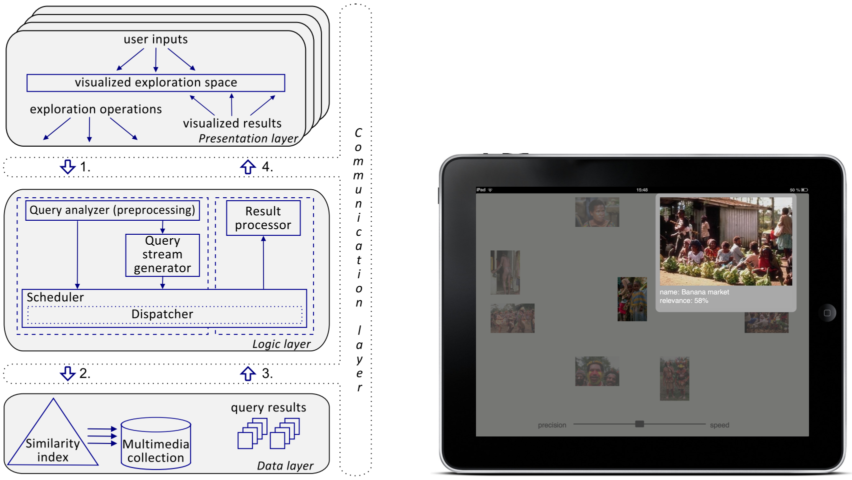
2.2 Real-Time Similarity Queries

Before we introduce our system, we provide necessary background for similarity queries performed in limited time frames to get results near real-time. The most popular queries are the k nearest neighbors (kNN) queries that return k most similar objects to the query object [21]. Our main intention for similarity exploration queries is to limit the response time. So, we define $k\text{NN}(t)$ as the timely limited kNN query which returns up to k most similar objects obtained from database objects accessed by the similarity index during query processing within the restricted time frame t. These queries occur in batches denoted as the *query stream* initiated as the result of user interaction with the system.

3 Implementing RTExp System

In this section, we provide the detailed information about the proposed **RTExp** (Real-Time Exploration) system and describe the data flow with typical use cases. The architecture consists of *presentation*, *logic*, and *data* layer with one extra layer which guarantees inter-layer communication and proper data flow between individual components (see Fig. 1a). The following list outline characteristics of individual layers:

1. The presentation layer represents intuitive and comfortable graphical user interface. Each user action to a visualized exploration space is transformed into the sequence of *exploration operations* and the corresponding query stream at the logic layer.



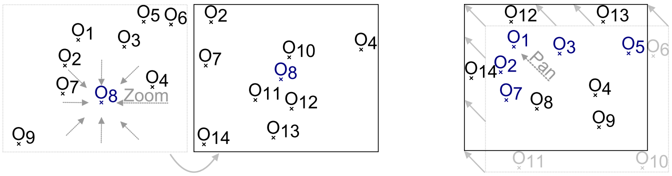
(a) High-level architecture **(b)** iPAD prototype

Fig. 1. RTExp System overview

2. The logic layer performs two main tasks: (1) translate user exploration operations into query streams, and (2) deliver partial/final results from the data layer back to the user. First, we decide whether (a) we terminate the currently running query stream, or (b) we create a new query stream (*Query stream generator*). The *Scheduler* schedules and controls all requests with a prioritized queue based on user settings, while the *Dispatcher* transfers query streams to be executed and the *Result processor* decides which query results are propagated to the presentation layer.
3. Data layer consists of the *multimedia collection* with a dedicated *similarity index*. We evaluate and compare several metric access methods (MAMs) [21], results are provided in Section 5. In the future, we also plan to involve non-metric access methods [19,1]. Two factors influence real-time capabilities and system performance:
   (a) *different query execution plans* for computationally *cheap* (executed ”as is”) and *expensive* (approximated in some way) similarity queries. For example, in M-tree [7] we use the top-down search strategy for cheap queries, however for the expensive ones we apply the bottom-up approach (fast traversing to the leaf level followed by enhancing the first results).
   (b) *instant evaluation* of similarity queries (*instant queries*). The benefits of instant queries can be utilized when (a) preferring quick results over the relevancy, (b) terminating obsolete queries, or (c) when a rapid increase in the number of simultaneous queries occurs.

4 Exploration Operations and User Interface

The more intuitive and ergonomic interface the client application provides, the more likely the end users will use the system and benefit from it. Inspired by the similarity-based layout approaches [14], we implement two most important user actions in the



(a) Zooming **(b)** Panning

Fig. 2. Exploration user actions

exploration process – *zooming* (Section 4.1) and *panning* (Section 4.2). We select these two actions as ideal candidates for basic exploration operations because they are intuitive and have been known to billions of Internet users as web mapping operations.

4.1 Zooming

Imagine we currently have a visualized set of items from the multimedia collection represented by objects O_i (see Fig. 2a). In this case, we are interested in the context of objects similar to the specific object O_8, so we *zoom in* to this object. The arrows show the zooming target which subsequently reveals previously invisible objects O_{10}, O_{11} $\ldots O_{14}$. The newly discovered objects are typically more similar than the previously shown objects or provide the visualization of more specific object clusters. On the contrary, if we apply the opposite action of *zooming out*, we get new objects that are less similar to the previous ones or we obtain more general object clusters.

Whenever a user executes a zooming action, we create the adequate exploration operation that consists of (a) the target zooming point, (b) the object closest to the target zooming point, and (c) objects closest to the boundaries. *Zoom in* initiates a single similarity query for which the query object is the closest object to the zooming target point, while for *zoom out* the query stream consists of multiple similarity queries for which the query objects are closest objects to the bounding rectangle.

4.2 Panning

This action advances the exploration in the specific direction (see Fig. 2b). The dotted rectangle represents the state before we apply *panning*, the solid one displays the new state, while the arrows outline the panning direction.

For panning actions, we create the appropriate exploration operation which includes several parameters such as (a) panning direction and its volume, (b) objects closest to visualized exploration space boundaries following the panning direction, and (c) objects that get outside the new visualized exploration space boundaries. Then, we initiate the query stream of similarity queries for which the query objects are objects closest to boundaries of the visualized exploration space following the panning direction.

4.3 RTExp Presentation Layer for Mobile Devices

To validate the feasibility of the proposed exploration system, we implemented the first prototype (called RTExp) with the user interface for iPad tablets (see Fig. 1b). The

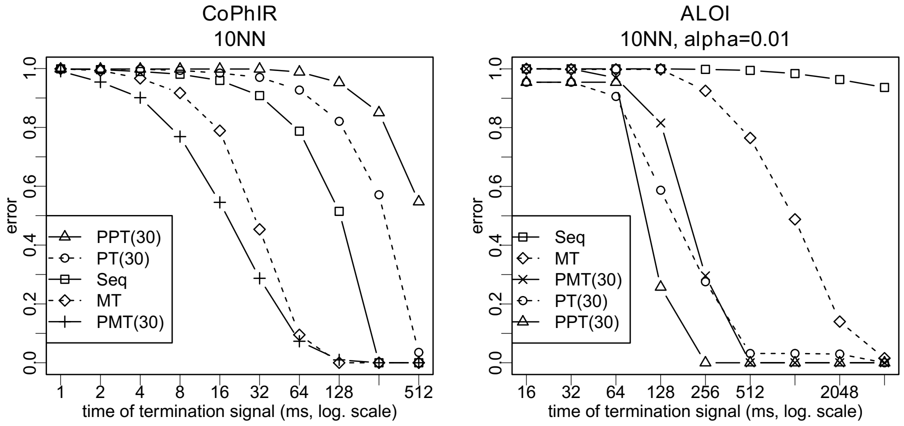


Fig. 3. *k*NN queries on the CoPhiR and ALOI database – inter-MAM comparison

powerful slider control at the bottom enables the user to adjust the exploration process. Sliding to the left gives more precise (but potentially slower) results, while putting the slider to the right returns faster (yet not necessary relevant) results.

5 Experimental Evaluations

Initially, we study the performance of RTExp system and depict the comparison of all evaluated MAMs: M-tree (*MT*), PM-tree (*PMT*), pivot tables (*PT*) and ptolemaic pivot tables (*PPT*) on two multimedia collections (see Fig. 3). While for subset of CoPhIR dataset [5], consisting of 1,000,000 images with cheap distance function, PM-tree is a clear winner (and pivot tables, behave even worse than the sequential scan (*Seq*)), the ALOI database [9], composed of 70,000 images with expensive SQFD [3], suits better for pivot tables. We also see that simple implementations of instant query termination may not be sufficient – the fastest index on ALOI reaches some reasonable results after 100ms. Hence, we need to design more specific approaches to instant query termination in the future.

Based on the previous experimental results, we employ instant query termination into simulation of real user *exploration scenario* (see Fig. 4). As the data layer, we use PM-tree, the best approach from the previous experiment. From results we can observe the major benefit of instant query termination - the constantly low execution time per a single exploration operation. Hence, the exploration process is really smooth, while the execution of the complete non-terminated operations would lead to unwanted delays.

Next, we study the precision error of exploration operations, each consists of several similarity queries. We join results from all (terminated) queries and compute the precision error with Jaccard distance [21] by comparing these results with joined results of complete (non-terminated) queries. The conclusion of this experiment is that the error correlates with the time of non-terminated query in case when all queries have the same amount of time for evaluation. Therefore, a *query analyzer* and/or modification of query evaluation is really necessary to improve the precision.

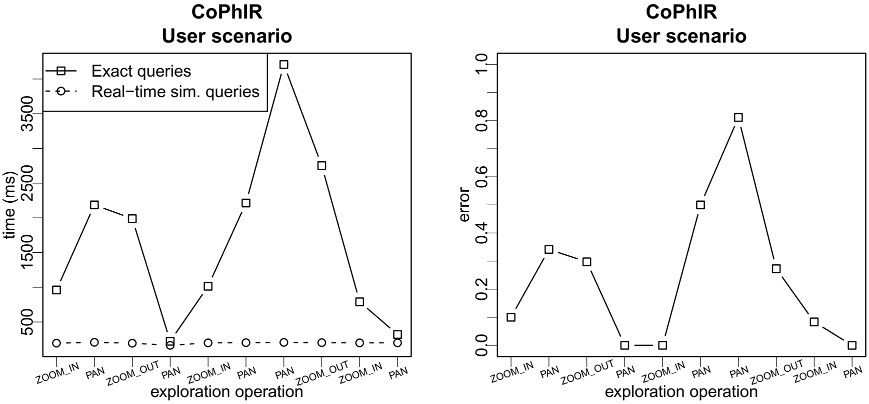


Fig. 4. User exploration scenario - time and precision error of exploration operations

6 Challenges and Future Work

During the development of RTExp system prototype, we experienced several issues that we would like to highlight, suggest possible solutions, and address them afterwards.

- **Continuous query evaluation.** For continuous performance, we intend to adapt the *publish/subscribe data delivery* [13]. Whenever a user executes an action, the visualized exploration space is continuously updated with partial/final results, while the system is evaluating the query stream in the background (*Result processor*).
- **Visualization of the multimedia collection.** We do not address the nontrivial problem of mapping the multimedia collection to the visualization space. We use the physical model based on the multidimensional scaling and the simulated annealing [8], successfully used and verified in another exploration system [12]. Later, we intend to employ more sophisticated solution such as the one suggested in [14].

Besides the already mentioned issues, in the future we want to focus on improving the precision of returned results while keeping the evaluation time at the same or even shorter values (e.g., employing data annotations).

7 Conclusion

Driven by requirements of end users, such as the modern user interface or truly continuous exploration, we propose the real-time multimedia exploration system RTExp that meets these criteria by providing scalable multi-layered architecture, applying real-time similarity exploration queries, and delivering intuitive user interface. Besides the description of the proposed system, we also verify our ideas in practice by developing a functional prototype with a user interface targeted for mobile devices and evaluating the system performance which demonstrates the viability of our multimedia exploration system.

Acknowledgments. This research has been supported in part by Czech Science Foundation project 202/11/0968 and by Grant Agency of Charles University projects 567312 and 910913.

References

1. Bartoš, T., Skopal, T., Moško, J.: Towards Efficient Indexing of Arbitrary Similarity. SIGMOD Record 42(2), 5–10 (2013)
2. Beecks, C., Skopal, T., Schöffmann, K., Seidl, T.: Towards large-scale multimedia exploration. In: DBRank 2011, Seattle, WA, USA, pp. 31–33 (2011)
3. Beecks, C., Uysal, M.S., Seidl, T.: Signature quadratic form distance. In: Conference on Image and Video Retrieval, CIVR 2010, pp. 438–445. ACM, New York (2010)
4. Böhm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM 33(3), 322–373 (2001)
5. Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., Rabitti, F.: CoPhIR: A test collection for content-based image retrieval. CoRR abs/0905.4627v2 (2009)
6. Chávez, E., Navarro, G.: A probabilistic spell for the curse of dimensionality. In: Buchsbaum, A.L., Snoeyink, J. (eds.) ALENEX 2001. LNCS, vol. 2153, pp. 147–160. Springer, Heidelberg (2001)
7. Ciaccia, P., Patella, M., Zezula, P.: M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In: VLDB 1997, pp. 426–435. Morgan Kaufmann Publishers Inc. (1997)
8. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Softw. Pract. Exper. 21(11), 1129–1164 (1991)
9. Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.: The amsterdam library of object images. International Journal of Computer Vision 61(1), 103–112 (2005)
10. Hjaltason, G.R., Samet, H.: Distance browsing in spatial databases. ACM Trans. Database Syst. 24(2), 265–318 (1999)
11. Hjaltason, G., Samet, H.: Incremental Similarity Search in Multimedia Databases. Computer science technical report series, University of Maryland (2000)
12. Lokoč, J., Grošup, T., Skopal, T.: Image exploration using online feature extraction and reranking. In: ICMR 2012, pp. 66:1–66:2. ACM, New York (2012)
13. McIlvride, B.: Nine core requirements for real-time cloud systems (January 2012), `http://real-timecloud.com/2012/01/12/nine-core-requirements-for-real-time-cloud-systems`
14. Nguyen, G.P., Worring, M.: Interactive access to large image collections using similarity-based visualization. Journal of Visual Languages and Computing 19(2), 203–224 (2008)
15. Patella, M., Ciaccia, P.: Approximate similarity search: A multi-faceted problem. J. of Discrete Algorithms 7(1), 36–48 (2009)
16. Santini, S., Jain, R.: Integrated browsing and querying for image databases. IEEE Multimedia 7(3), 26–39 (2000)
17. Schaefer, G.: A next generation browsing environment for large image repositories. Multimedia Tools and Applications 47, 105–120 (2010), doi:10.1007/s11042-009-0409-2
18. Schoeffmann, K., Ahlstrom, D., Beecks, C.: 3D Image Browsing on Mobile Devices. In: ISM 2011, pp. 335–336. IEEE Computer Society, Washington, DC (2011)
19. Skopal, T.: Unified framework for fast exact and approximate search in dissimilarity spaces. ACM Trans. Database Syst. 32(4) (November 2007)
20. Uhlmann, J.K.: Implementing Metric Trees to Satisfy General Proximity/Similarity Queries (1991) manuscript
21. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach. Springer (2005)
22. Zezula, P., Savino, P., Amato, G., Rabitti, F.: Approximate similarity retrieval with m-trees. The VLDB Journal 7(4), 275–293 (1998)